数理解析研究所講究録
1068 巻 1998 年 194-202

A REINFORCED SURROGATE CONSTRAINTS METHOD FOR SEPARABLE NONLINEAR INTEGER PROGRAMMING

Yuji NAKAGAWA (仲川 勇二)

Kansai University, Faculty of Informatics,

Abstract

This paper provides a new reinforcement to the surrogate constraints method for solving separable nonlinear integer programming problems with a few constraints. The surrogate constraints method often has a duality gap, i.e., fails to find an exact solution to the original problem. A reinforcement proposed to fill the gap is to solve a sequence of target problems that enumerate all solutions hitting a target region with a single constraint.

1. INTRODUCTION

This paper presents a surrogate constraints method reinforced for solving a separable nonlinear integer problem with multiple constraints. The surrogate constraints method solves a sequence of surrogate problems, which have a single constraint, instead of a primal multi-constrained problem. However, surrogate constraints method often has a duality gap, i.e., fails to produce an exact optimal solution of the primal problem. The present reinforcement can fill the surrogate duality gaps when there exist gaps.

Consider the following separable nonlinear integer programming problem.

$$[\mathrm{P}^{\mathrm{O}}]:\quad \text{Maximize}\quad f^{\mathrm{O}}(\mathbf{x}) = \sum_{i=1}^{n} f_i^{\mathrm{O}}(x_i)$$

$$\text{subject to}\quad g_j^{\mathrm{O}}(\mathbf{x}) = \sum_{i=1}^{n} g_{ji}^{\mathrm{O}}(x_i) \le b_j^{\mathrm{O}} \quad \text{for } j = 1, \ldots, m, \qquad (1)$$

$$x_i \in K_i^{\mathrm{O}} \quad \text{for } i = 1, \ldots, n,$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, $K_i^{\mathrm{O}} = \{1, 2, \ldots, k_i^{\mathrm{O}}\}$ and , without loss of generality, we assume that

$$\begin{array}{l} f_i^{\mathrm{O}}(x_i) \ge 0 \quad \text{for } x_i = 1, \ldots, k_i^{\mathrm{O}}, \quad i = 1, \ldots, n \\ g_{ji}^{\mathrm{O}}(x_i) \ge 0 \quad \text{for } x_i = 1, \ldots, k_i^{\mathrm{O}}, \quad i = 1, \ldots, n, \quad j = 1, \ldots, m \end{array} \qquad (2)$$

The problem $[\mathrm{P}^{\mathrm{O}}]$ is called Multidimensional Nonlinear Knapsack Problem (MNKP) by Morin and Marsten(1976). As several special cases, the MNKP includes 1) Nonlinear Resource Allocation Problem, named by Bretthauer and Shetty(1995), has differentiable convex objective and constraint functions. 2) Resource Allocation Problem, e.g. Ibaraki and Katoh(1988), has the convex objective function and the single constraint of the sum of variables. 3)Multiple Choice Knapsack Problem presented by Nauss(1978) is the linearization of problem $[\mathrm{P}^{\mathrm{O}}]$ with a single constraint.

Surrogate constraints are introduced into mathematical programming by Glover(1968). Luenberger(1968) showed that any quasi-convex programming problems can be solved exactly if the surrogate multipliers are correctly chosen. Karwan and Rardin(1984) give some empirical evidences on the effectiveness of surrogate constraints in integer linear programming. There are several algorithms for choosing an optimal surrogate multiplier vector to MNKP. Dyer(1980) proposed two algorithms; one analogous to generalized linear programming and the other to subgradient method. Nakagawa et al.(1984) present Cutting-Off Polyhedron (COP) algorithm that generates a sequence of multiplier vectors by cutting off polyhedrons and using a center of the polyhedron as a next multiplier vector.

Most of problems $[\mathrm{P}^{\mathrm{O}}]$ include surrogate duality gaps unfortunately. In other words, solving a surrogate problem with an optimal surrogate multiplier vectors, surrogate constraints method fails to produce an exact solution of $[\mathrm{P}^{\mathrm{O}}]$. A reinforced method, which we call Slicing algorithm, is proposed to find an exact solution of $[\mathrm{P}^{\mathrm{O}}]$ in the feasible region of the optimal surrogate problem. Slicing algorithm solves a sequence of target problems that enumerate all solutions in a slice of the feasible region. Sometimes we have a difficulty of solving target problems with too wide a target solution space. The slicing method uses two practical techniques to reduce the difficulty. One is to thin out the target solution space and the other is to narrow the feasible region.

2. SURROGATE DUAL

The surrogate problem $[\mathrm{P}^{\mathrm{S}}(\mathbf{u})]$ corresponding to $[\mathrm{P}^{\mathrm{O}}]$ are written as follows:

$$
\begin{aligned}
[\mathrm{P}^{\mathrm{S}}(\mathbf{u})]:\ \ & \text{Maximize} && f^{\mathrm{O}}(\mathbf{x}) \\
& \text{subject to} && \mathbf{u}\mathbf{g}^{\mathrm{O}}(\mathbf{x}) \le \mathbf{u}\mathbf{b}^{\mathrm{O}}, \\
& && \mathbf{x} \in \mathbf{K}^{\mathrm{O}},
\end{aligned}
$$

where

$$
\begin{aligned}
&\mathbf{u} = (u_1, u_2, \ldots, u_m) \in R^m, \\
&\mathbf{g}^{\mathrm{O}}(\mathbf{x}) = (g_1^{\mathrm{O}}(\mathbf{x}),\ g_2^{\mathrm{O}}(\mathbf{x}), \ldots, g_m^{\mathrm{O}}(\mathbf{x}))^t, \\
&\mathbf{b}^{\mathrm{O}} = (b_1^{\mathrm{O}},\ b_2^{\mathrm{O}}, \ldots, b_m^{\mathrm{O}})^t \\
&\mathbf{K}^{\mathrm{O}} = \left\{\mathbf{x} : x_i \in K_i^{\mathrm{O}} \ \textit{for}\ i = 1, 2, \ldots, n\right\}.
\end{aligned}
$$

The surrogate dual problem is defined by

$$
[\mathrm{P}^{\mathrm{SD}}]:\ \ \min\left\{v^{\mathrm{OPT}}[\mathrm{P}^{\mathrm{S}}(\mathbf{u})] : \mathbf{u} \in \mathbf{U}\right\}
$$

where $v^{\mathrm{OPT}}[\bullet]$ means the optimal objective function value of problem $[\bullet]$ and

$$
\mathbf{U} = \left\{\mathbf{u} \in R^m : \sum_{j=1}^{m} u_j = 1, \mathbf{u} \ge \mathbf{0}\right\}.
$$

The surrogate problem $[\mathrm{P}^{\mathrm{S}}(\mathbf{u})]$ has the following property.

Property 1: Let $\mathbf{x}^q$ be an optimal solution to problem $\mathrm{P}^{\mathrm{S}}(\mathbf{u}^q)$ for a surrogate multiplier vector $\mathbf{u}^q \in \mathbf{U}$. For any $\mathbf{u} \in \mathbf{U}$ such that $\mathbf{u}\mathbf{g}^{\mathrm{O}}(\mathbf{x}^q) > \mathbf{u}\mathbf{b}^{\mathrm{O}}$, it is held that

$$v^{\mathrm{OPT}}[\mathrm{P}^{\mathrm{S}}(\mathbf{u})] \geq f(\mathbf{x}^q).$$

This property means that the region $\{\mathbf{u} \in \mathbf{U} : \mathbf{u}\mathbf{g}^{\mathrm{O}}(\mathbf{x}^q) > \mathbf{u}\mathbf{b}^{\mathrm{O}}\}$ can be removed from $\mathbf{U}$ as the price of obtaining $\mathbf{x}^q$. An algorithm presented by Dyer (1980) or Nakagawa et al. (1984) can generate a sequence of multiplier vectors $\mathbf{u}^1, \mathbf{u}^2, \ldots, \mathbf{u}^q$ that covers the whole of $\mathbf{U}$ in the meaning of Property 1. The multiplier vector $\mathbf{u}^*$ such that

$$v^{\mathrm{OPT}}[\mathrm{P}^{\mathrm{S}}(\mathbf{u}^*)] = \min\{v^{\mathrm{OPT}}[\mathrm{P}^{\mathrm{S}}(\mathbf{u}^1)], v^{\mathrm{OPT}}[\mathrm{P}^{\mathrm{S}}(\mathbf{u}^2)], \ldots, v^{\mathrm{OPT}}[\mathrm{P}^{\mathrm{S}}(\mathbf{u}^q)]\}$$

is optimal to the surrogate dual $[\mathrm{P}^{\mathrm{SD}}]$. If an optimal solution $\mathbf{x}^{\mathrm{SD}}$ to the problem $[\mathrm{P}^{\mathrm{S}}(\mathbf{u}^*)]$, i.e. $[\mathrm{P}^{\mathrm{SD}}]$,is feasible to the primal problem $[\mathrm{P}^{\mathrm{O}}]$, then $\mathbf{x}^{\mathrm{SD}}$ is an exact optimal solution of $[\mathrm{P}^{\mathrm{O}}]$. When the sequence provides no feasible solutions to $[\mathrm{P}^{\mathrm{O}}]$, it is said that there exists a surrogate duality gap. Then the value $f^{\mathrm{O}}(\mathbf{x}^{\mathrm{SD}})$ provides an upper bound on the optimal objective function value of $[\mathrm{P}^{\mathrm{O}}]$.

3. RESOLUTION OF SURROGATE DUALITY GAP

In order to fill the surrogate duality gap, consider the following problem.

$$
\begin{array}{lll}
[\mathrm{P}^{\mathrm{T}}(f^{\mathrm{T}}, b^{\mathrm{T}})]: & \multicolumn{2}{l}{\text{Enumerate all solutions } \mathbf{x} \text{ hitting}} \\
 & \text{a target} & f^{\mathrm{O}}(\mathbf{x}) \geq f^{\mathrm{T}} \\
 & \text{subject to} & \mathbf{u}^*\mathbf{g}^{\mathrm{O}}(\mathbf{x}) \leq b^{\mathrm{T}}, \\
 & & \mathbf{x} \in \mathbf{K}^{\mathrm{O}}.
\end{array}
$$

where $\mathbf{u}^*$ is an optimal multiplier vector to $[\mathrm{P}^{\mathrm{SD}}]$ corresponding to a primal problem $[\mathrm{P}^{\mathrm{O}}]$. Miyaji et al. (1995) calls this problem target problem. If $f^T \leq v^{\mathrm{OPT}}[\mathrm{P}^{\mathrm{O}}]$, then a target problem

$[\mathrm{P}^{\mathrm{T}}(f^{\mathrm{T}}, \mathbf{u}^* \mathbf{b})]$ can enumerates exact optimal solutions to $[\mathrm{P}^{\mathrm{O}}]$. The target values f^T will be chosen from an interval such as $f^{\mathrm{O}}(\mathbf{x}^{\mathrm{Near}}) \le f^T \le f^{\mathrm{O}}(\mathbf{x}^{\mathrm{SD}})$, where $\mathbf{x}^{\mathrm{Near}}$ is a near optimal solution to $[\mathrm{P}^{\mathrm{O}}]$. The solutions hitting the target are called target solutions. MA without dominance testing can solve exactly the target problem $[\mathrm{P}^{\mathrm{T}}(f^{\mathrm{T}}, \mathbf{u}^* \mathbf{b})]$. The problem $[\mathrm{P}^{\mathrm{T}}(f^{\mathrm{T}}, \mathbf{u}^* \mathbf{b})]$ becomes harder to solve with decreasing value of f^T because of increasing number of target solutions.

Consider a sequence of problems $[\mathrm{P}^{\mathrm{T}}(f^{\mathrm{T}}, \mathbf{u}^* \mathbf{b})]$ with $f^{\mathrm{T}} = \varphi_0, \varphi_1, \ldots, \varphi_\kappa$ where

$$\varphi_t = \frac{(\kappa - t) f^{\mathrm{O}}(\mathbf{x}^{\mathrm{SD}}) - t f^{\mathrm{O}}(\mathbf{x}^{Near})}{\kappa} \quad (t = 0, 1, 2, \ldots, \kappa).$$

These target problems are solved by Slicing algorithm in the order of $[\mathrm{P}^{\mathrm{T}}(\varphi_0, \mathbf{u}^* \mathbf{b})], \ldots, [\mathrm{P}^{\mathrm{T}}(\varphi_\kappa, \mathbf{u}^* \mathbf{b})]$, as shown in Figure 1. When Slicing algorithm finds out an optimal solution to $[\mathrm{P}^{\mathrm{O}}]$ out of target solutions, the algorithm stops. However, the problems become much harder to solve with increasing problem size. The difficulties are divided into two cases. The first case of difficulty is that there exist too many target solutions with the same objective function value. The second case is that the feasible region satisfying the constraint is too wide. In Slicing algorithm, two techniques are used to decrease the difficulties. One technique is to thin out the target solutions. When the number of alternative items for a variable exceeds a threshold value, a thinning out law in Slicing algorithm is practiced. The other technique against the difficulties is to slice off one piece after another of its feasible region by changing values of b^{T}. It should be noted that an optimal solution obtained by using these techniques may not be guaranteed to be exact optimal to the primal problem $[\mathrm{P}^{\mathrm{O}}]$.

Consider a sequence of problems $[\mathrm{P}^{\mathrm{T}}(\varphi_k, b^{\mathrm{T}})]$ with $b^{\mathrm{T}} = \beta_0, \beta_1, \ldots, \beta_\nu$ where

$$\beta_t = \frac{(\nu - t)\mathbf{u}^* \mathbf{g}^{\mathrm{O}}(\mathbf{x}^{\mathrm{Near}}) - t\mathbf{u}^* \mathbf{b}^{\mathrm{O}}}{\nu} \quad (t = 0, 1, 2, \ldots, \nu).$$

Slicing algorithm tries to solve the problems $[\mathrm{P}^{\mathrm{T}}(\varphi_k, \beta_t)]$ in the order of $t = 0, 1, \ldots, \nu$. Figure 2 illustrates this technique.

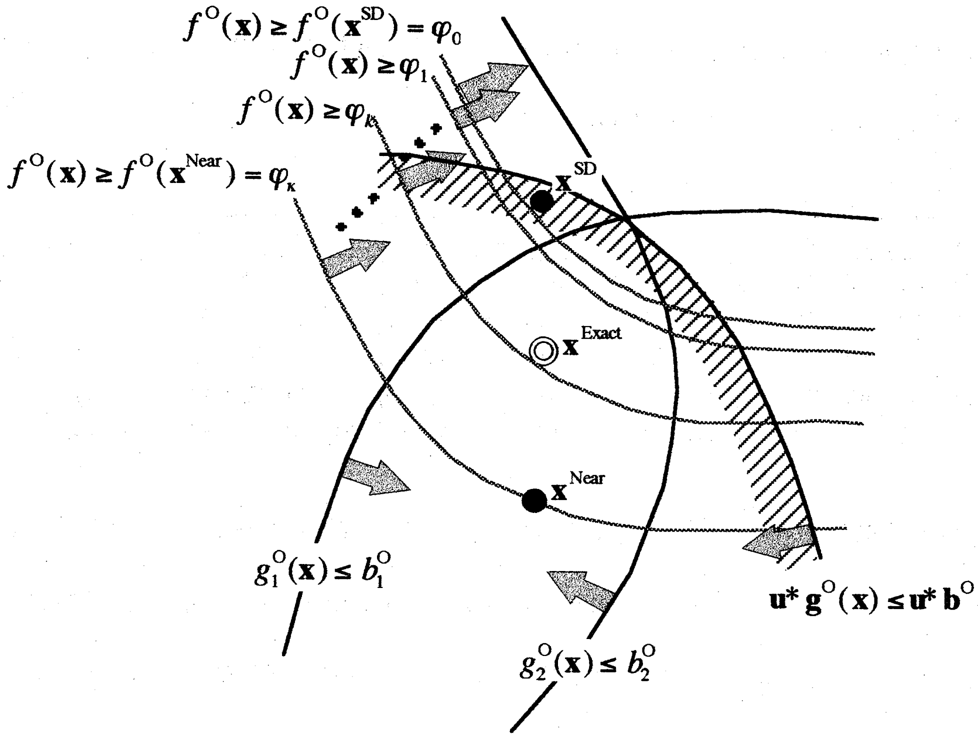

Fig. 1

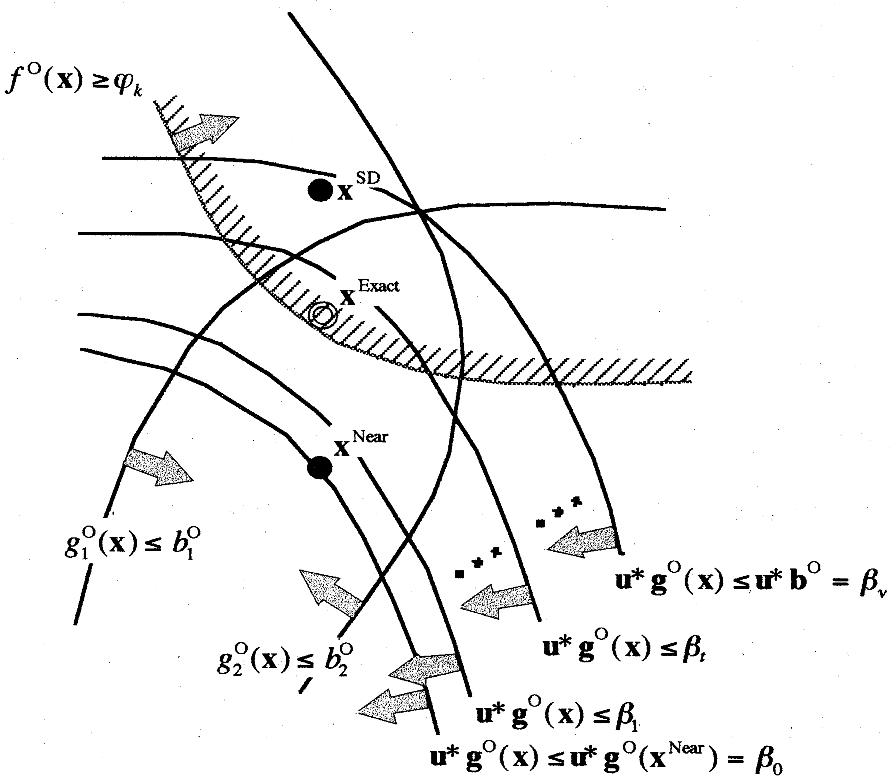

Fig. 2

4. EXAMPLE

Slicing algorithm will be demonstrated on the following simple problem with three constraints.

This problem has $m=3$, $n=5$, $k_i^{\mathrm{O}}=4$ for $i=1,\dots,5$, and $\mathbf{b}^{\mathrm{O}}=(234,188,191)^t$. The values of functions $f_i^{\mathrm{O}}(x_i)$ and $g_{ji}^{\mathrm{O}}(x_i)$ are as follows:

Table 1 Example

k	1	2	3	4
$f_1^{\mathrm{O}}(k)$	29	35	47	71
$f_2^{\mathrm{O}}(k)$	21	52	74	85
$f_3^{\mathrm{O}}(k)$	5	12	36	66
$f_4^{\mathrm{O}}(k)$	30	33	47	70
$f_5^{\mathrm{O}}(k)$	28	51	58	70

k	1	2	3	4
$g_{11}^{\mathrm{O}}(k)$	8	40	63	88
$g_{12}^{\mathrm{O}}(k)$	28	54	74	83
$g_{13}^{\mathrm{O}}(k)$	24	55	57	70
$g_{14}^{\mathrm{O}}(k)$	11	38	68	91
$g_{15}^{\mathrm{O}}(k)$	15	31	47	51

k	1	2	3	4
$g_{21}^{\mathrm{O}}(k)$	20	26	35	51
$g_{22}^{\mathrm{O}}(k)$	1	21	22	35
$g_{23}^{\mathrm{O}}(k)$	6	21	31	43
$g_{24}^{\mathrm{O}}(k)$	23	50	73	96
$g_{25}^{\mathrm{O}}(k)$	20	48	76	81

k	1	2	3	4
$g_{31}^{\mathrm{O}}(k)$	20	34	45	69
$g_{32}^{\mathrm{O}}(k)$	15	17	29	36
$g_{33}^{\mathrm{O}}(k)$	12	29	61	73
$g_{34}^{\mathrm{O}}(k)$	27	55	58	81
$g_{35}^{\mathrm{O}}(k)$	4	13	27	46

Cutting-Off Polyhedron (COP) algorithm presented by Nakagawa et. al 1984 is started with $\mathbf{u}^1=(\frac{1}{3},\frac{1}{3},\frac{1}{3})$, as shown in Fig. 2. The surrogate subproblem is $[\mathrm{P}^{\mathrm{S}}(\mathbf{u}^1)]$: Max $f^{\mathrm{O}}(\mathbf{x})$ s.t. $\mathbf{u}^1\mathbf{g}^{\mathrm{O}}(\mathbf{x})\le\mathbf{u}^1\mathbf{b}^{\mathrm{O}}$ and $\mathbf{x}\in\mathbf{K}^{\mathrm{O}}$. The computer code based on MA produces an optimal solution $\mathbf{x}^1$ =(1, 3, 4, 1, 4), $f^{\mathrm{O}}(\mathbf{x}^1)$=269.0. We have the first cutting plane $-24u_1-3u_2>-4$ from $\mathbf{u}\mathbf{g}^{\mathrm{O}}(\mathbf{x}^1)>\mathbf{u}\mathbf{b}^{\mathrm{O}}$ and $u_3=1-u_1-u_2$. COP algorithm generates $\mathbf{u}^2=(0.0536,0.4881,0.4583)$ as a center of balance of a material points system that has an unit weight at all vertices. Similarly the surrogate constraints method generates $\mathbf{x}^2$=(4,3,4,1,1), $f^{\mathrm{O}}(\mathbf{x}^2)$=269, $-13u_1-40u_2>-11$, $\mathbf{u}^3$ =(0.0734, 0.1478, 0.7787), $\mathbf{x}^3$=(1, 4, 4, 1, 3), $f^{\mathrm{O}}(\mathbf{x}^3)=268$, and the last cutting plane $-7u_1-17u_2>-8$. This plane cuts off the remaining region of U as shown in Fig. 3. Therefore the optimal surrogate multiplier is $\mathbf{u}^*=\mathbf{u}^3$, the optimal solution of surrogate dual problem is $\mathbf{x}^{\mathrm{SD}}=\mathbf{x}^3$ and $f^{\mathrm{O}}(\mathbf{x}^{\mathrm{SD}})$=268. However, $\mathbf{x}^3$ is infeasible to the second constraint $g_2^{\mathrm{O}}(\mathbf{x})\le b_2^{\mathrm{O}}$. There exists a surrogate duality gap. In order to fill this surrogate gap, consider target problems $[\mathrm{P}^{\mathrm{T}}(f^{\mathrm{T}},b^{\mathrm{T}})]$ of $b^{\mathrm{T}}=\mathbf{u}^*\mathbf{b}^{\mathrm{O}}$=193.7 and f^{T}= 268, 267, …, 261, where a near optimal solution $\mathbf{x}^{\mathrm{Near}}$=(1,4,4,1,2), $f^{\mathrm{O}}(\mathbf{x}^{\mathrm{Near}})$=261 which obtained by a heuristic method. The sequence of the target problems is solved by MA without dominance testing. The problems $[\mathrm{P}^{\mathrm{T}}(268,193.7)]$, . .., $[\mathrm{P}^{\mathrm{T}}(262,193.7)]$ have no target solutions that is feasible to the primal problem. The problem $[\mathrm{P}^{\mathrm{T}}(261,193.7)]$ provides a target solution $\mathbf{x}^{\mathrm{Slice}}$=(1, 4, 4, 1, 2) satisfying all constraints of the primal problem. The solution is the exact optimal solution to the primal problem.

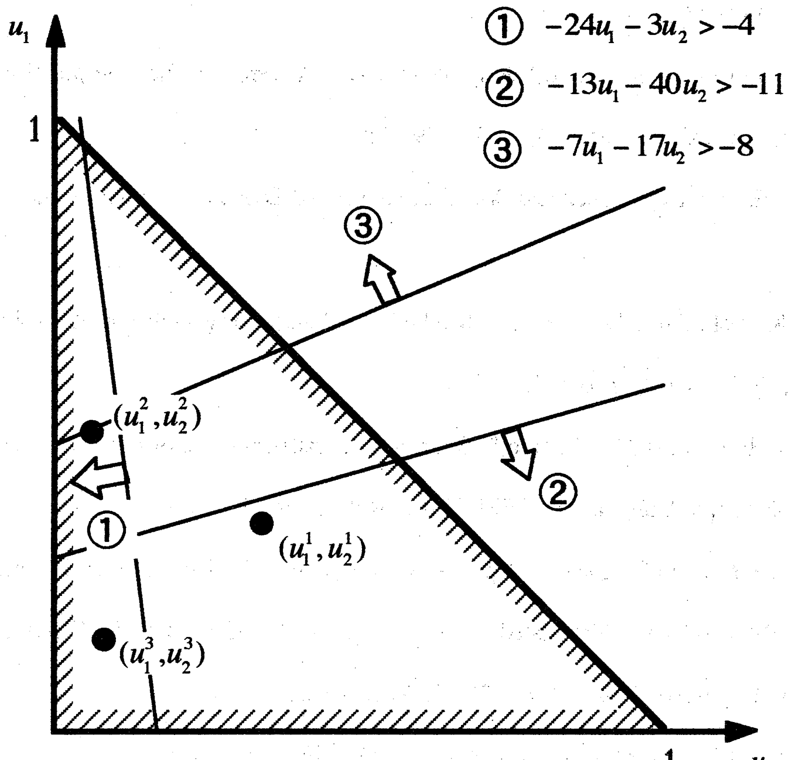

Fig. 3

References

Armstrong, R. D., D. S. Kung, P. Sinha and A. A. Zoltners, "A Computational Study of a Multiple-Choice Knapsack Algorithm," ACM Trans. on Math. Software, 9 (1983), 184-198.

Dyer, M. E.,"Calculating Surrogate Constraints," Math. Programming, 19 (1980), 255-278.

Dyer, M. E., N. Kayal and J. Walker,"A Branch and Bound Algorithm for Solving The Multiple-Choice Knapsack Problem," J. Computational and Applied Math., 11 (1984), 231-249.

Glover, F., "Surrogate Constraints," Oper. Res., 16 (1968) 741-749.

Ibaraki, T,"Enumerative Approaches to Combinatorial Optimization," Annals of Oper. Res., 10 (1987), 1-602.

Ibaraki, T. and N. Katoh, Resource Allocation Problems, MIT Press, Cambridge, Mass (1988).

Karwan, M. H. and R L. Radin, "Surrogate Dual Multiplier Search Procedures in Integer Programming," Oper. Res. 32 (1984) 52-69.

Luenberger, D. G., "Quasi-convex programming", SIAM J. Applied Math., 16 (1968) 1090-1095.

Marse, K. and S. D. Roberts,"Implementing a Portable FORTRAN Uniform (0,1) Generator," Simulation, 41 (1983), 135-139.

Marsten, R. E. and T. L. Morin,"A Hybrid Approach to Discrete Mathematical Programming," Math.

Programming, 14 (1977), 21-40.

Miyaji, I., Y. Nakagawa and K. Ohno,"Decision support system for the composition of the examination problem," European J. Oper. Res., 80(1995), 139-138.

Morin, T. L. and R. E. Marsten,"Branch-and-Bound Strategies for Dynamic Programming," Oper. Res., 24(1976), 611-627.

Nakagawa, Y. and K. Nakashima,"A Heuristic Method for Determining Optimal Reliability Allocation," IEEE Trans. Reliab., R-26 (1977), 156-161.

Nakagawa, Y., M. Hikita and H. Kamada," Surrogate Constraints Algorithm for Reliability Optimization Problems with Multiple Constraints," IEEE Trans. Reliab., R-33 (1984) 301-305.

Nakagawa, Y. "A New Method for Discrete Optimization Problems," Electronics and Communications in Japan, Part 3, 73 (1990) 99-106. Translated from Trans. of the Institute of Electronics, Information and Communication Engineers, 73-A (1990), 550-556 (In Japanese).

Nakagawa, Y. , Easy C Programming, Asakurashoten (1996) (in Japanese).

Nauss, R. M, "The 0-1 Knapsack Problem with Multiple Choice Constraints," European J. of Oper. Res., 2 (1978) 125-131.

Ohtagaki, H., Y. Nakagawa, A. Iwasaki, and H. Narihisa,"Smart Greedy Procedure for Solving a Nonlinear Knapsack Class of Reliability Optimization Problems," Math. Comput. Modeling, 22 (1995) 261-272.

Sinha, P. and A. Zoltners,"The Multiple-Choice Knapsack Problem," Oper. Res., 27 (1978), 125-131.